TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

———————— * ————————

BÁO CÁO BÀI TẬP LỚN

Môn: Hệ gợi ý

Đề tài:

Xây dựng hệ gợi ý phim cho người dùng

Nhóm thực hiện: Nhóm 9

Lê Hải Nam – 20173264
Lê Anh Thành – 20170115
Dương Kiều Nga – 20173280
Trịnh Thị Phương - 20152916

Lớp: **IT4613**

Giảng viên hướng dẫn: **ThS. Ngô Văn Linh**

Hà Nội, tháng 12 năm 2020

MỤC LỤC

1. Tổng quan bài toán thực tế

Ngày nay, dịch vụ xem phim trực tuyến đang ngày càng được ưa chuộng, phát triển do nhu cầu tiện ích của con người. Nhiều các trang web, nền tảng cung cấp dịch vụ phim trực tuyến cũng vì thế tăng lên, cũng như loại phim và số lượng bộ phim ngày càng nhiều. Điều đó khiến cho người dùng gặp khó khăn trong việc lựa chọn phim để xem. Từ đó yêu cầu các nền tảng cần có một tiện ích hỗ trợ cho người dùng dễ dàng hơn trong việc chọn lựa phim phù hợp với sở thích của mình, để có thể thu hút được lượt truy cập, tăng doanh thu và lợi nhuận.

=> Hệ gợi ý phim là một giải pháp hữu ích, hiệu quả

- Ưu điểm của hệ gợi ý:
  - Tiện lợi cho người dùng.
  - Đơn giản cho người dùng. Không cần người dùng trả lời câu hỏi hay khảo sát.
  - Tạo cảm giác gây nghiện xem phim cho người dùng nếu gợi ý phù hợp.

- Thách thức đối với hệ gợi ý phim:
  - Số lượng phim rất nhiều.
  - Đa dạng, nhiều chủng loại khác biệt.
  - Sở thích cá nhân đa dạng.
  - Các đánh giá của người dùng đôi khi còn "khan hiếm".
- Bài toán giải quyết: Xây dựng được một phương pháp gợi ý phim cho người dùng hiệu quả, giải quyết đối với trường hợp dữ liệu đáng giá của người dùng khan hiếm (xử lý cho dữ liệu implicit feedbacks)

2. Tập dữ liệu

Tập dữ liệu sẽ sử dụng là tập dữ liệu từ trang Movielens cung cấp bởi GroupLens

2.1. Tổng quan về bộ dữ liệu movielens

Tập dữ liệu mô tả các xếp hạng của user theo thang đánh giá từ 0.5 sao – 5 sao thu thập từ trang web Movielens- một dịch vụ phim trực tuyến. Bộ dữ liệu được GroupLens- một nhóm nghiên cứu thuộc Khoa Khoa học và Kỹ thuật Máy tính tại Đại học Minnesota tổng hợp và cung cấp.

Dữ liệu được chứa trong các tập tin genome-scores, genome-tags, links, movies, ratings và tags. Bộ dữ liệu này và các bộ dữ liệu củaGroupLens khác có sẵn để tải xuống công khai tại http://grouplens.org/datasets/

- Cấu trúc tệp dữ liệu ratings

- Chứa thông tin dữ liệu đánh giá xếp hạng của người. Mỗi dòng của tệp này sau hàng tiêu đề thể hiện một xếp hạng cho một bộ phim của một người dùng và có định dạng sau: ***userId, movieId, rating, timestamp***
- Các dòng trong tệp này được sắp xếp đầu tiên theo userId, sau đó, với mỗi user, theo movieId.
- Xếp hạng được thực hiện trên thang điểm 5 sao, với gia số 0.5 sao (0,5 sao - 5,0 sao) - trừ các bộ xuất hiện trước bộ ml-10M100K.

- Cấu trúc tệp dữ liệu tags

- Tất cả các thẻ được gán bởi người dùng chứa trong tệp tags. Mỗi dòng của tệp này sau hàng tiêu đề đại diện cho một thẻ được một người dùng gán cho một bộ phim và có định dạng sau: ***userId, movieId, tag, timestamp***
- Thẻ là siêu dữ liệu do người dùng tạo về phim. Mỗi thẻ thường là một từ đơn hoặc cụm từ ngắn. Ý nghĩa, giá trị và mục đích của một thẻ cụ thể được xác định bởi mỗi người dùng.

- Cấu trúc tệp dữ liệu movies

- Thông tin phim được chứa trong tệp này. Mỗi dòng của tệp sau hàng tiêu đề đại diện cho thông tin của một bộ phim và có định dạng sau: ***movieId, title, genres***
- Tên phim được nhập thủ công hoặc nhập từ https://www.themoviedb.org/ và bao gồm năm phát hành.

- Cấu trúc tệp dữ liệu links

- Các định danh có thể được sử dụng để liên kết đến các nguồn dữ liệu phim khác được chứa trong tệp này. Mỗi dòng của tệp sau hàng tiêu đề đại diện cho một bộ phim và có định dạng sau: ***movieId, imdbId, tmdbId***

- Tag Genome (genome-score và genome-tags)

- Tập dữ liệu này bao gồm bản sao hiện tại của Bộ gen thẻ. Bộ gen thẻ là một cấu trúc dữ liệu chứa điểm số liên quan của thẻ đối với phim.
- Cấu trúc là một ma trận dày đặc: mỗi đoạn phim trong bộ gen có giá trị cho mọi thẻ trong bộ gen. Bộ gen thẻ mã hóa mức độ mạnh mẽ của các bộ phim thể hiện các đặc tính cụ thể được thể hiện bằng các thẻ (không khí, kích thích tư duy, thực tế, v.v.). Bộ gen thẻ được tính bằng thuật toán máy học trên nội dung do người dùng đóng góp bao gồm thẻ, xếp hạng và đánh giá văn bản.
- Bộ gen được chia thành hai tệp. Tệp genome-scores chứa dữ liệu về mức độ liên quan của thẻ phim ở định dạng sau: ***movieId , tagId, relevance***
- Tệp thứ hai genome-tags, cung cấp mô tả thẻ cho các ID thẻ trong tệp bộ gen, ở định dạng sau: ***tagId, tag***

2.2. Dữ liệu sử dụng

Ở đây sẽ dựa vào dữ liệu từ đánh giá của người dùng từ đó đưa qua mô hình để gợi ý cho người dùng những item tương ứng phù hợp với sở thích. Dữ liệu sử dụng sẽ lấy từ tệp ratings.

	userId	movieId	rating	timestamp
0	1	2	3.5	1112486027
1	1	29	3.5	1112484676
2	1	32	3.5	1112484819
3	1	47	3.5	1112484727
4	1	50	3.5	1112484580

Hình 1. Định dạng tệp ratings

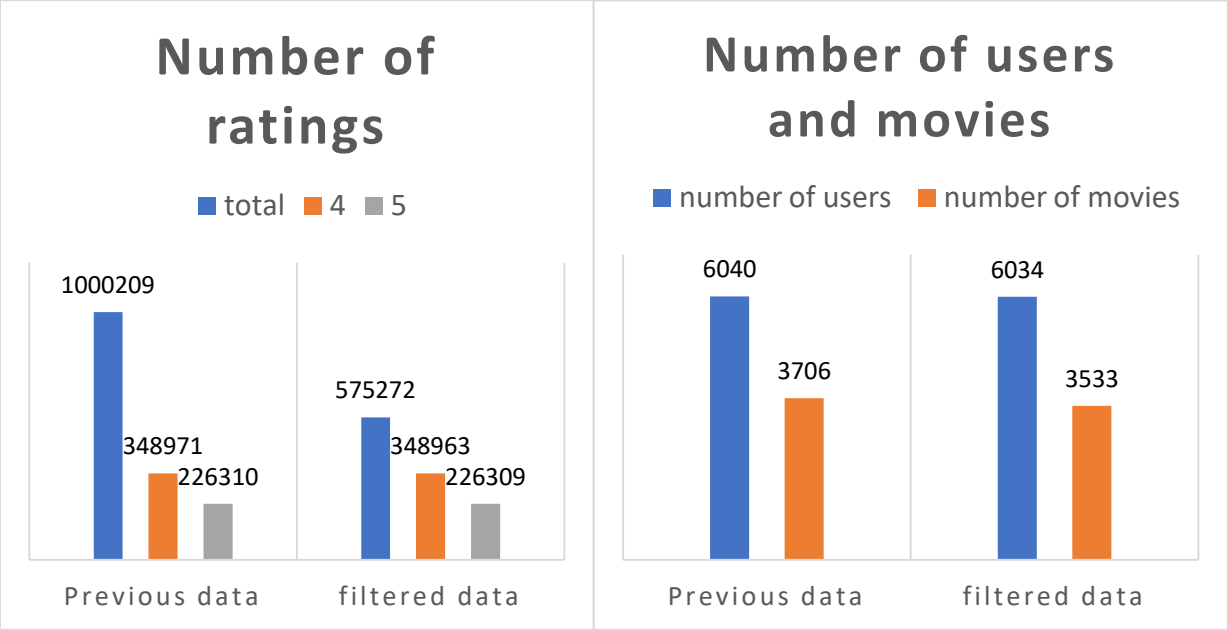
- Dữ liệu gồm 4 cột : userId, movieId, rating, timestamp trong đó:
  - mỗi userId tương ứng một user khác nhau đánh giá movie trên hệ thống
  - movieId tương ứng với ID của bộ phim trên hệ thống.
  - rating(0.5 sao – 5.0 sao) là đánh giá của userId đối với movieId

Bốn tập ml-1m, ml-10M100K, ml-20m và ml-25m được sử dụng để huấn luyện và đánh giá thử nghiệm phương pháp.

- Mô tả dữ liệu:
  - Dữ liệu ban đầu: Dữ liệu ratings từ Movielens
  - Dữ liệu sau: Dữ liệu sau chi bỏ đi các rating 3.5, giữ lại các userId có số lượt đánh giá trên 5 phim.

- Các trường:
  - số movieId: số lượng phim trong tập dữ liệu
  - số userId: số lượng user trong tập dữ liệu
  - số rating: xếp hạng đánh giá của user

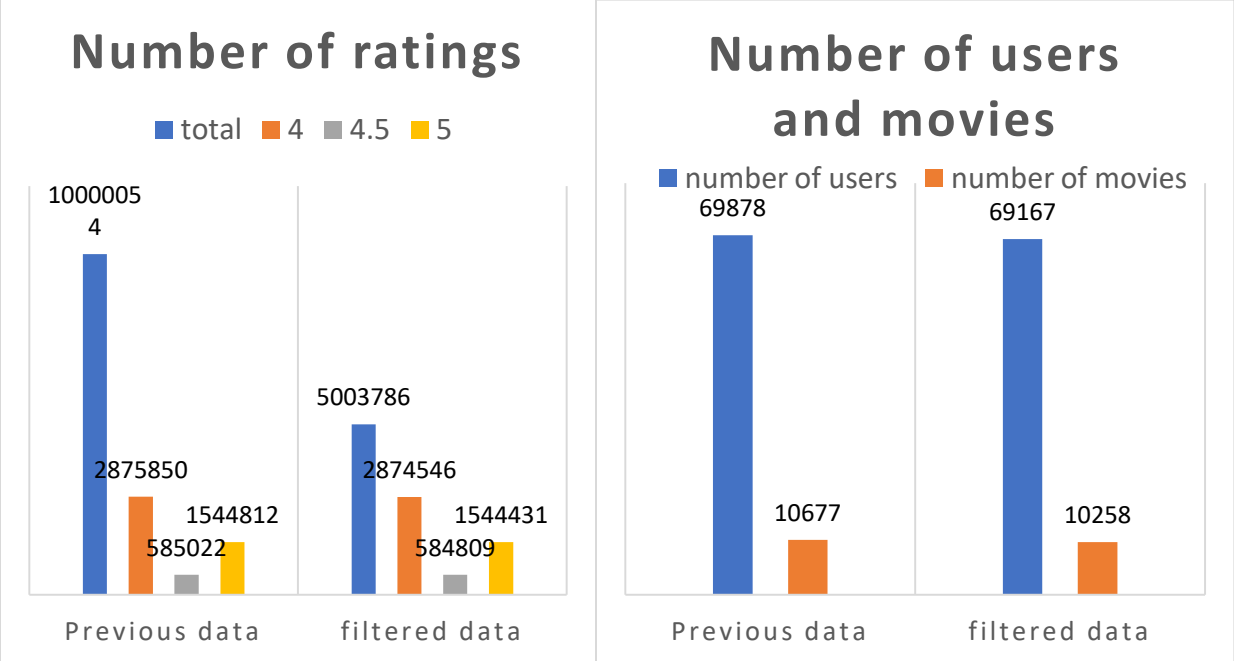
- Bộ ml-1m:



Sparsity: 97.3 %

Hình 2. Thống kê user, item, rating bộ ml-1m

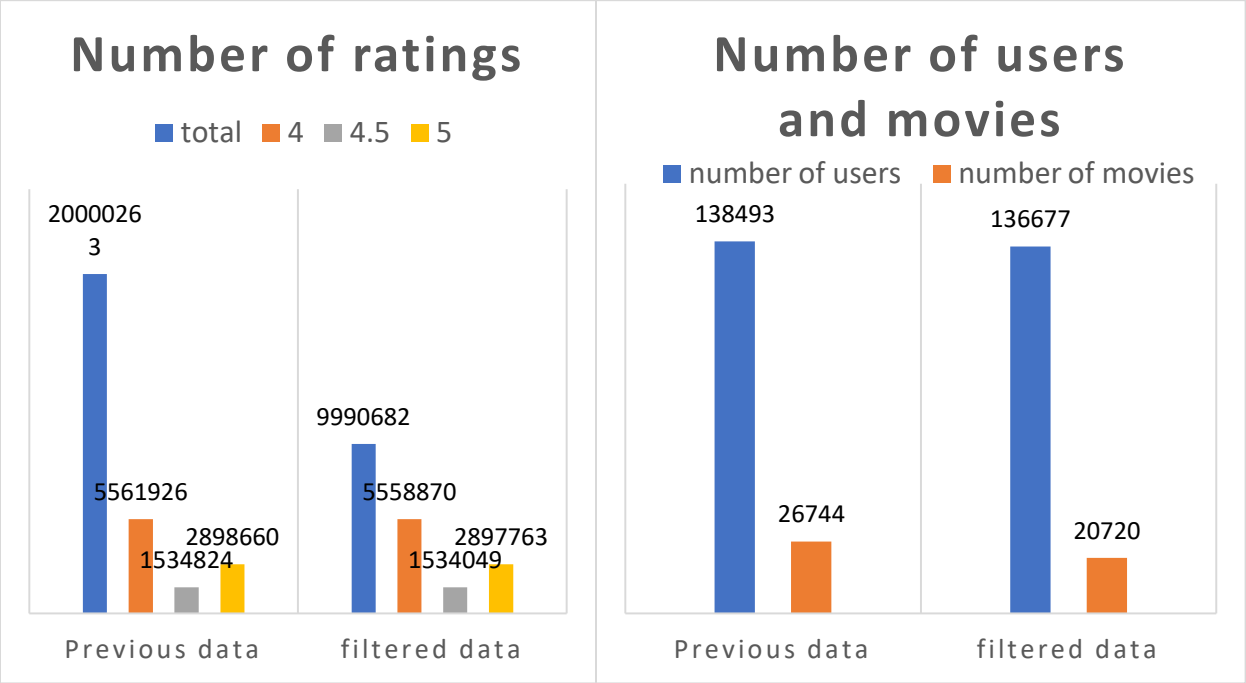
- Bộ ml-10M100K:



Sparsity: 99.3 %

Hình 3. Thống kê user, item, rating bộ ml-10M100K

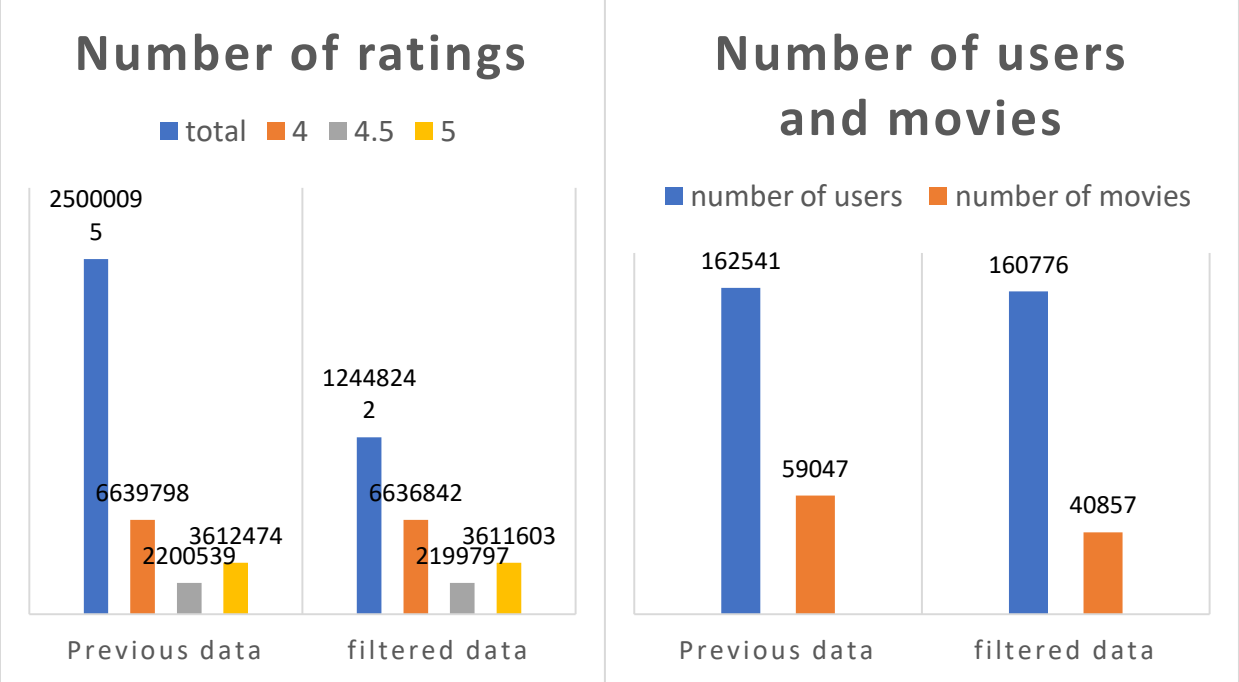
- Bộ ml-20m:



Sparsity: 99.65 %

Hình 4. Thống kê user, item, rating bộ ml-20m

- Bộ ml-25m:



Sparsity: 99.81 %

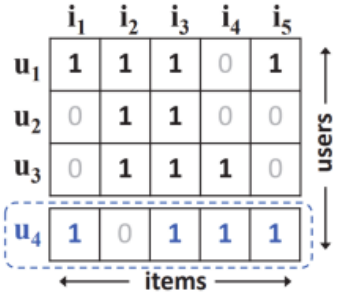
Hình 5. Thống kê user, item, rating bộ ml-25m

3. Phương pháp giải quyết

3.1. Xử lý dữ liệu

Loại dữ liệu bài toán hướng tới giải quyết là kiểu dữ liệu 'implicit feedbacks', nhưng tập dữ liệu movielens chỉ cung cấp dữ liệu explicit- các ratings của người dùng. Do đó, ta sẽ nhị phân hóa dữ liệu ratings của người dùng, thể hiện là người dùng đã tương tác với một bộ phim nào đó.

Lúc đó ma trận users – items là một ma trận nhị phân (x_{ij} = 0 – người dùng i chưa đánh giá phim j ngược lại x_{ij} = 1- người dùng i đã đánh giá phim j).



Hình 6. Ma trận users – item nhị phân

Đối với dữ liệu implicit, chúng ta khó có thể biết được người dùng thực sự có thích bộ phim đó hay không, do hành động chỉ là 'click' hoặc 'play'. Các rating thấp (1 sao, 2 sao) thể hiện rõ việc người dùng không thích bộ phim đó, do đó ta sẽ giữ lại các đánh giá phim trên 3.5 sao để có thể phù hợp với đặc điểm của dạng dữ liệu implicit feedbacks và loại các user đánh giá phim ít (dưới 5 phim).

Mỗi bộ dữ liệu sau đó sẽ được chia ra làm ba bộ train, valid và test để thực hiện học mô hình và đánh giá kết quả trên từng bộ dữ liệu.

3.2. Lọc cộng tác (Collaborative Filtering)

Phương pháp lọc cộng tác là phương pháp phân tích dữ liệu người dùng để tìm ra mối tương quan giữa các đối tượng người dùng. Lọc cộng tác hoạt động bằng cách xây dựng một cơ sở dữ liệu, lưu trữ dưới dạng ma trận người dùng (users) - sản phẩm (items).

- Là các tiếp cận nổi bật để đưa ra gợi ý:

- Được sử dụng bởi nhiều các hệ thống thương mại điện tử.
- Có nhiều thuật toán và các biến thể.
- Áp dụng trong nhiều lĩnh vực gợi ý (news, book, movies, DVDs, ..)

- Giả thiết và ý tưởng:

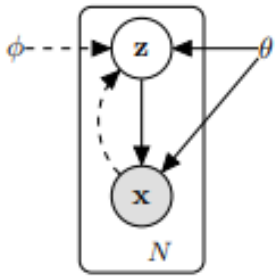
- Users đưa ra đánh giá cho các items (implicit hoặc explicit feedback)
- Users có sở thích tương tự nhau trong quá khứ, cũng sẽ có sở thích tương tự trong tương lai.

- Các phương pháp lọc cộng tác cơ bản

- Lọc cộng tác dựa trên láng giềng
- Lọc cộng tác dựa trên mô hình
- …

3.3. Mô hình Variational Auto-Encoder (VAE)

Variational Auto- Encoder (VAE) là một mô hình sinh có khả năng học biểu diễn của dữ liệu. VAE là phương pháp thuộc nhóm suy diễn "variational inference", học ra các thành phần ẩn có khả năng tổng quát hóa các mô hình tuyến tính và cho phép khám phá các mô hình xác suất phi tuyến tính.



Hình 7. Sơ đồ mô hình, nguồn [2]

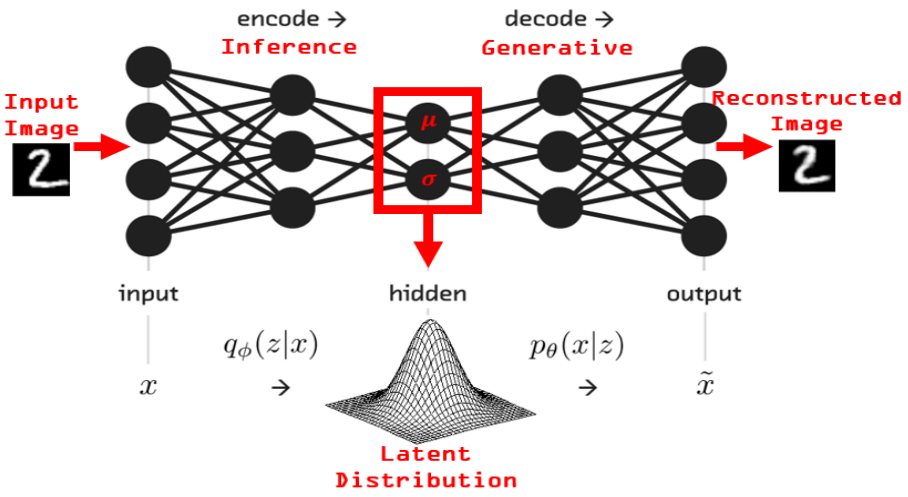
z: lớp ẩn để học sinh biểu diễn cho dữ liệu x. Prior: z ~ P(z | θ)

Mô hình gồm 2 phần chính là Encoder- mã hóa dữ liệu thành một biểu diễn ẩn và Decoder- giải mã biểu diễn ẩn để sinh lại dữ liệu ban đầu.

- Encoder (bộ mã hóa): Dữ liệu x đưa qua phép biến đổi g(x, φ) như một phép mã hóa để tạo ra thành phần ẩn z: z ~ q(. | g(x, φ))
- Decoder (bộ giải mã): Biểu diễn ẩn z đưa qua phép biến đổi f(z, θ) để sinh lại biểu diễn của dữ liệu: x ~ p(. | f(z, θ))
- Hàm mục tiêu: $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[-\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) + \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\right]$

Do việc học đều được biểu diễn dưới dạng phân phối xác suất nên biến ngẫu nhiên ẩn z có khả năng tổng quát hóa sinh lại được ra các dữ liệu thuộc cùng miền phân phối với các dữ liệu đầu vào đại diện. Các hàm g và f được sử dụng là các mạng neuro với φ và θ lần lượt là tham số của mạng, để có thể khái quát được phân phối phức tạp.

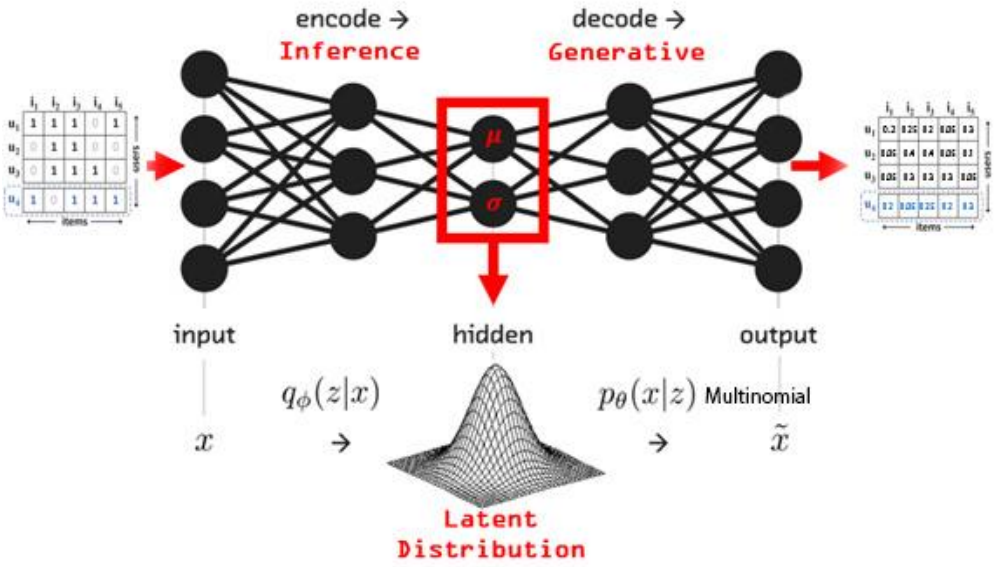
Luồng thực hiện của phương pháp được mô tả chi tiết ở hình 8.



Hình 8. Nguồn internet

3.4. Áp dụng VAE vào Collaborative Filtering

Phương pháp áp dụng giải quyết bài toán gợi ý phim được sử dụng mở rộng mô hình VAE, kết hợp với phương pháp lọc cộng tác. Đầu vào là ma trận users – items, đầu ra là phần trăm khả năng yêu thích trên mỗi bộ phim của người dùng; từ đó đưa ra top-k bộ phim với tỉ lệ cao cho người dùng để gợi ý.



Hình 9.

- Mô hình hóa:

  - Tập users: u = { u_1, u_2. . . , u_u }. Tập items: I = {i_1, i_2 . . . , i_i}
  - Ma trận users – items: x = {x_{ij} | i = 1..U, j = 1…I}

với $\begin{cases} xij = 0 nếu u_i chưa tương tác với item i_j \\ xij = 1 nếu u_i đã tương tác với item i_j \end{cases}$ (như hình 6)

- Quá trình sinh của mô hình:

  - Đối với mỗi user u, mô hình bắt đầu bằng lấy mẫu biểu diễn ẩn k chiều z_u từ phân phối chuẩn Gaussian(0,1)- Phân phối tiên nghiệm của biến ngẫu nhiên z.
  - Biểu diễn ẩn z_u được biến đổi thông qua một hàm phi tuyến tính $f(\cdot \mid \theta) \in R^I$ để tạo ra một phân phối xác suất trên I items $\pi(z_u)$

$$\mathbf{z}_u \sim \mathcal{N}(0, \mathbf{I}_K), \quad \pi(\mathbf{z}_u) \propto \exp\{f_\theta(\mathbf{z}_u)\}, \qquad \text{với } N_u = \sum_i^I x_{ui}$$
$$\mathbf{x}_u \sim \text{Mult}(N_u, \pi(\mathbf{z}_u)).$$

  - Từ biến ẩn khái quát z ta có thể sinh lại biểu diễn users – items qua likelihood $p_\theta(x_u|z_u$.Phân phối của likelihood được chọn là phân phối Multinomial, qua thực nghiệm cho thấy phân phối này phù hợp với dữ liệu click hơn các phân phối như Gausian hay logistic likelihood.

$$log p_\theta(x_u|z_u \cong \sum_i x_{ui} log \pi_i(z_u)$$

- Chọn 'variational distribution' : $q_\varphi(z_u|x_u) = N\left(\mu_\varphi(x_u), \sigma_\varphi(x_u)\right)$

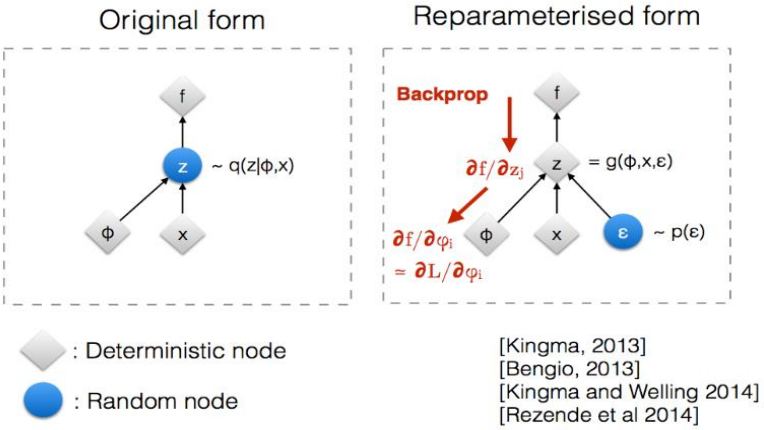
- Hàm mục tiêu (ELBO):

$$\log p(\mathbf{x}_u; \theta) \geq \mathbb{E}_{q_\phi(\mathbf{z}_u \mid \mathbf{x}_u)}[\log p_\theta(\mathbf{x}_u \mid \mathbf{z}_u)] - \text{KL}(q_\phi(\mathbf{z}_u \mid \mathbf{x}_u) \| p(\mathbf{z}_u))$$
$$\equiv \mathcal{L}(\mathbf{x}_u; \theta, \phi)$$

- KL- reweighting: Đại lượng (kullback leibler divergence) KL có thể coi như một đại lượng 'regularization'. Nó giúp ràng buộc đại lượng phân phối hậu nghiệm với tiền nghiệm trong quá trình huấn luyện. Việc đặt hệ số cho đại lượng này sẽ giúp ta kiểm soát trong việc đánh đổi việc học phù hợp với dữ liệu hay ràng buộc phân phối hậu nghiệm với tiền nghiệm.

$$\mathcal{L}_\beta(\mathbf{x}_u; \theta, \phi) \equiv \mathbb{E}_{q_\phi(\mathbf{z}_u \mid \mathbf{x}_u)}[\log p_\theta(\mathbf{x}_u \mid \mathbf{z}_u)]$$
$$-\beta \cdot \text{KL}(q_\phi(\mathbf{z}_u \mid \mathbf{x}_u) \| p(\mathbf{z}_u)).$$

- Reparameterization trick



Hình 10. Reparameterization trick

Việc đạo hàm ngược sẽ bị tắc ở nút lấy mẫu z, đồng thời việc lấy mẫu z trực tiếp từ μvàσ dẫn đến 'high variance'. Vì vậy biến ngẫu nhiên được biến đổi z = g(x, φ, ∈) với ∈ là một biến hỗ trợ ∈ ~ p(∈).

Từ đó giảm 'high variance' và giải quyết được vấn đề đạo hàm ngược.

- Thuật toán:

Algorithm 1: VAE-SGD Training collaborative filtering VAE with stochastic gradient descent.

Input: Click matrix $\mathbf{X} \in \mathbb{R}^{U \times I}$
Randomly initialize θ, ϕ
while *not converged* **do**
  Sample a batch of users $\mathcal{U}$
  forall $u \in \mathcal{U}$ **do**
    Sample $\epsilon \sim \mathcal{N}(0, \mathbf{I}_K)$ and compute $\mathbf{z}_u$ via reparametrization trick
    Compute noisy gradient $\nabla_\theta \mathcal{L}$ and $\nabla_\phi \mathcal{L}$ with $\mathbf{z}_u$
  end
  Average noisy gradients from batch
  Update θ and ϕ by taking stochastic gradient steps
end
return θ, ϕ

Hình 11. Giải thuật huấn luyện mô hình VAE kết hợp lọc cộng tác, nguồn [1]

4. Cài đặt thử nghiệm

4.1. Công cụ sử dụng

- Ngôn ngữ lập trình: python3 – ngôn ngữ phổ biến, hữu ích để lập trình học máy
- IDE: Google Colab – tận dụng tốc độ của GPU
- Thư viện: pytorch, numpy, matplotlib.

4.2. Cài đặt thử nghiệm

- Chia dữ liệu: Dữ liệu được chia ra làm 3 tập train, valid và test. Tập valid và test lại được chia ra làm 2 tập tỉ lệ 2/8 dựa vào các items user đã đánh giá- 1 tập dùng để đưa qua mạng với các item ở tập còn lại coi như là chưa biết, tập còn lại là tập chứa các item đã đánh giá của người dùng nhưng được coi như là chưa biết để thử nghiệm.

- ml-1m: train- 5034 users, valid- 500 users, test- 500 user
- ml-10M100K: 59167 users, valid- 5000 users, test- 5000 users
- ml-20m: train- 116677 users, valid- 10000 users, test- 10000 users
- ml-25m: train- 130776 users, valid- 15000 users, test- 15000 users

- Mô hình gồm tổng 3 tầng ẩn: input → [600 200 600] → output
- Nhân tố ẩn z có số chiều là 200 và 2 tầng 600 tương ứng với tầng ẩn của mạng encode và decode.

- Quá trình huấn luyện sử dụng giải thuật Adam để tối ưu, tham số tốt nhất thu được với:
  - ml-20m, ml25m: tốc độ học bằng 0.001, mini batch gồm 500 dữ liệu, huấn luyện với 200 epochs.
  - ml-10M100K: tốc độ học bằng 0.001, mini batch gồm 100 dữ liệu, huấn luyện với 200 epochs.
  - ml-1m: tốc độ học 0.0001, mini batch gồm 100 dữ liệu, huấn luyện với 200 epochs.
  - Giá trị Dropout cho đầu vào là 0.5; sử dụng KL-reweighting với hệ số β động.
- Tham số β tốt nhất cho các bộ: ml-1m (0.1), ml-10M100K (0.16), ml-20m (0.17), ml-25m (0.2).
- Code cài đặt được lưu trong tệp *VAE_4CF.ipynb* đi kèm với báo cáo. Code nhóm cài đặt được tham khảo từ nguồn: https://github.com/dawenl/vae_cf

4.3. Kết quả thử nghiệm

4.3.1. Độ đo đánh giá

Sử dụng 2 độ đo là Recall và nDCG

Đó là các phương pháp sử dụng đánh giá trong việc truy xuất thông tin.

- Recall: là độ đo dùng để đo tính đầy đủ, xác định tỉ lệ phần có liên quan được truy xuất trên tổng số có liên quan.

Độ đo này được xây dựng dựa trên Ground Truth được thiết lập bởi chuyên gia. Trong đó đối với hệ gợi ý:

- True Positive (tp): sản phẩm được gợi ý và trong thực tế là tốt.
- False Positive (fp): sản phẩm được gợi ý và trong thực tế là không tốt.
- False Negative (fn): sản phẩm không được gợi ý và trong thực tế là tốt.
- True Negative (tn): sản phẩm không được gợi ý và trong thực tế là không tốt.

Và vì thế, trong hệ gợi ý, Recall chính là độ đo xác định tỉ lệ số sản phẩm được gợi ý trên số sản phẩm tốt (cần được gợi ý).

$$Recall = \frac{tp}{tp + fn} = sảnphẩmđượcgợiý \vee \frac{}{sảnphẩmcầnđượcgợiý \vee}$$

- nDCG (Normalized Discounted Comulative Gain):

Recall là một độ đo tốt, tuy nhiên đối với một hệ gợi ý, thì độ đo Recall chưa phản ánh được hết vai trò của các sản phẩm được gợi ý khi coi chúng như nhau. Trong thực tế, thông thường chúng ta sẽ gợi ý một danh sách dạng top-N, khi đó nếu các sản phẩm cần được gợi ý đứng ở vị trí càng cao trong danh sách thì kết quả gợi ý đó sẽ càng tốt.

Vì thế ta cần thêm một độ đo khác để có thể đánh giá được dựa trên cả thứ tự gợi ý sản phẩm, và nDCG là một độ đo như vậy.

nDCG là một giá trị được chuẩn hóa từ hai giá trị là DCG (Discounted Comulative Gain) là độ đo sử dụng hệ số giảm logarit để tính giá trị cho kết quả gợi ý dựa trên các vị trí mà các sản phẩm được đề xuất và IDCG (Idealized Discounted Comulative Gain) là độ đo tương tự DCG nhưng trong trường hợp lí tưởng, đó là tất cả các sản phẩm được đề xuất đúng nằm trên đầu danh sách gợi ý. nDCG được tính bằng tỉ lệ giữa DCG và IDCG.

Độ đo DCG được tính bằng công thức:

$$DCG_{pos} = rel_1 + \sum_{i=2}^{pos} \frac{rel_i}{log_2 i}$$

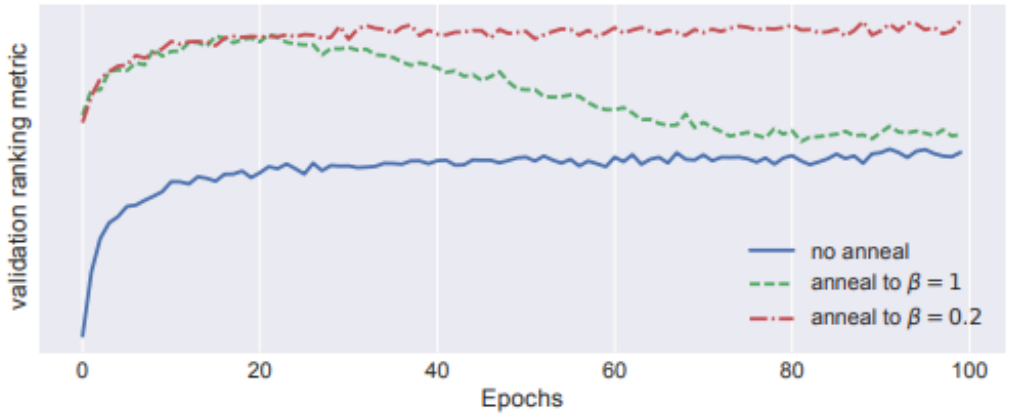
Trong đó: *pos* chính là vị trí của các gợi ý được xét, reli chính là mức độ liên quan của sản phẩm được gợi ý tại vị trí thứ *i* trong danh sách gợi ý, giá trị này bằng 1 nếu như gợi ý đó là chính xác và bằng 0 trong trường hợp còn lại. Hệ số $log_2 i$ chính là hệ số giảm logarit được sử dụng.

Độ đo IDCG được tính bằng công thức:
$IDCG_{pos} = rel_1 + \sum_{i=2}^{|h|-1} \frac{rel_i}{log_2 i}$ Trong đó: *h* là tập các sản phẩm được gợi ý chính xác. Độ đo này chính là *DCG* trong trường hợp lí tưởng, đó là khi tất cả các sản phẩm gợi ý đúng đều nằm trên đầu danh sách gợi ý.

nDCG chính bằng tỉ lệ giữa DCG và IDCG, và trở thành giá trị được chuẩn hóa trong đoạn [0,1]. Và vì có khả năng đánh giá được cả mức độ gợi ý có tốt không dựa trên thứ tự danh sách được gợi ý nên độ đo này thường được sử dụng để đánh giá độ tốt của một hệ gợi ý.

4.3.2. Kết quả thử nghiệm

- Việc lựa chọn KL- reweighting với hệ số β động tham khảo bài báo của phương pháp này, do kết quả thực nghiệm cho thấy kết quả hơn đáng kể so với việc sử dụng hệ số tĩnh hoặc không sử dụng.



Hình 12: Validation ranking metrics với cách 'set up' hệ số β khác nhau, nguồn [1]

- Trong bài báo gốc việc so sánh thử nghiệm cũng được thực hiện trên 2 bộ dữ liệu về phim (Netflix và ml-20m) với các phương pháp khác thể hiện rõ phương pháp VAE cho lọc cộng tác vượt trội hơn các phương pháp trước đó cho hệ gợi ý phim đối với dữ liệu 'implicit'.

(a) ML-20M

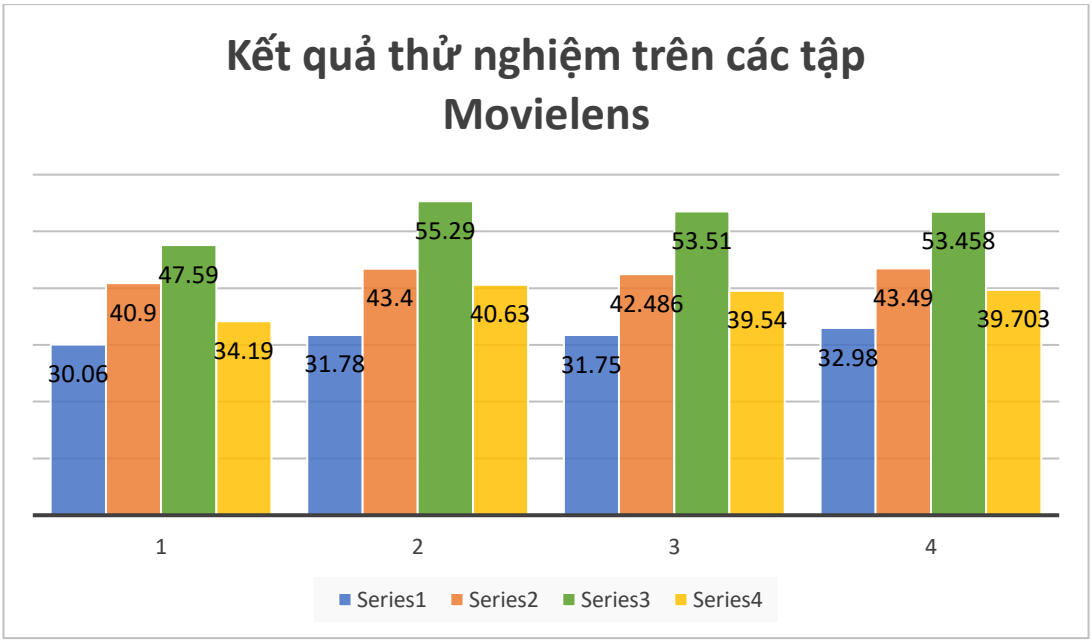
	Recall@20	Recall@50	NDCG@100
Mult-VAE^{PR}	**0.395**	**0.537**	**0.426**
WMF	0.360	0.498	0.386
SLIM	0.370	0.495	0.401
CDAE	0.391	0.523	0.418

(b) Netflix

	Recall@20	Recall@50	NDCG@100
Mult-VAE^{PR}	**0.351**	**0.444**	**0.386**
WMF	0.360	0.498	0.386
SLIM	0.370	0.495	0.401
CDAE	0.391	0.523	0.418

Hình 13. Kết quả thử nghiệm mô hình trên Netflix và ml-20m, nguồn [1]

- Nhóm đã thực hiện thử nghiệm thêm trên 3 bộ dữ liệu là ml-1m, ml-10M100K và ml-25m. Kết quả đánh giá trên các tập test mô tả trên **Hình 14**, kết quả thể hiện sự ổn định của phương pháp khi độ đo trên các tập là khá tương đồng nhau.



Hình 14

		NDCG@10	NDCG@100	Recall@10	Recall@50
AE	ml-1m	32.04	42.33	31.57	48.47
	ml-10M100K	32.57	43.45	40.48	54.66
	ml-20m	31.72	41.89	38.77	52.32
	ml-25m	33.013	43.1	39.003	52.37
VAE	ml-1m	30.06	40.9	34.19	47.59
	ml-10M100K	31.78	43.4	40.63	55.29
	ml-20m	31.75	42.486	39.54	53.51
	ml-25m	32.98	43.49	39.703	53.46

Hình 15. So sáng việc sử dụng AutoEncoder và Variational Autoencoder

5. Tổng kết

Nhóm đã đưa ra một phương pháp giúp giải quyết bài toán gợi ý phim ảnh cho người dùng. Phương pháp vẫn có khả năng khái quát cao học được biểu diễn của dữ liệu.

Phương pháp đã giải quyết được vấn đề nêu ra, đó chính là xử lý trên thông tin 'implicit'; việc làm việc tốt trên kiểu thông tin này sẽ giúp cho các nền tảng không phải lo lắng về cách thu thập đánh giá rõ ràng từ người dùng mà vẫn có thể gợi ý phim cho người dùng.

Tuy nhiên phương pháp là thuần túy kết hợp sử dụng Lọc cộng tác nên vẫn còn mặt hạn chế về trường hợp 'cold start'- khi một user hay một bộ phim mới xuất hiện thì sẽ gợi ý như thế nào?

Trong tương lai, nhóm sẽ phát triển bằng cách kết hợp các phương pháp khác như Lọc dựa trên nội dung để giải quyết vấn đề 'cold start' hay áp dụng thêm sự liên hệ giữa các items với nhau để gợi ý. Việc xấp xỉ phân phối của q(z) còn đơn giản – 'mean field' nên sẽ cải tiến bằng những phương pháp phức tạp hơn để q(z) có thể bắt được nhiều đặc trưng của dữ liệu.

Tài liệu tham khảo

[1]. Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, Tony Jebara, 2018, Variational Autoencoders for Collaborative Filtering, arXiv:1802.05814 (2018).

[2]. Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013).

[3]. Koren et al. 2009, Matrix factorization techniques for recommender systems, Computer 42 (2009), no. 8, 30–37.